In the Claims:

1. (Currently Amended) A method for computing distances between a received point for a tone received in a receiver and four points in a two-dimensional grid with a constellation representing a number of bits greater than three, ~~wherein each of the four points belong to a unique coset in the constellation,~~ the method comprising:

determining a first point on a grid nearest to the received point;

computing a second point closest to the received point inside a specified ~~area;~~ area after determining the first point;

computing a third, a fourth, and a fifth point, after computing the second point, wherein each of the second, the third, the fourth and the fifth [[point]] points is a member of a different coset in the constellation and ~~each point~~ is the closest point in its coset to the received point; and

computing ~~a distance~~ distances from the received point to each of the second, the third, the fourth, and the fifth ~~points.~~ points; and

providing the computed distances to a decoder.

2. (Currently Amended) The method of claim 1 further ~~comprising~~ comprising, after the ~~first~~ computing the second point, recomputing the second point if the second point is invalid.

3. (Original) The method of claim 2, wherein the second point is invalid if it is outside of the constellation.

4. (Currently Amended) The method of claim 1, wherein the first point can be determined by evaluating:

~~round((Rx + iRy – 1 – i)/2*2 + 1 + I,~~ round((Rx + iRy – 1 – i)/2)*2 + 1 + I,

wherein Rx and Ry are two-dimensional components of the received point, [[i]] I is [[the]] an imaginary number, and round(.) is an operator that returns an integer number closest to a value provided to it.

5. (Currently Amended) The method of claim 1, wherein the number of bits is an even value, wherein the received point can be expressed in two-dimensional components Rx and Ry, and wherein the ~~first~~ computing the second point comprises:

determining if Rx and Ry lie inside a square specified by the number of bits; and

computing two-dimensional components of the second point based on ~~the second determining~~ whether Rx and Ry lie inside the square.

6. (Currently Amended) The method of claim 5, wherein the ~~second determining~~ computing the second point further comprises:

setting Cx = 1 if Rx lies inside a boundary of the square, else Cx = -1;

setting Cy = 1 if Ry lies inside [[a]] the boundary of the square, else Cy = -1;

~~and wherein the fourth computing comprises~~

[[ ]]setting Ax = sign(RGx) * MAX_{XY} if Cx = -1, else Ax = RGx; and

[[ ]]setting Ay = sign(RGy) * MAX_{XY} if Cy = -1, else Ay = RGy,

wherein Ax and Ay are two-dimensional components of the second point, RGx and RGy are two-dimensional components of the first point, and MAX_{XY} is a value describing [[the]] a size of the square and can be computed by $2^{\text{number of bits/2}} - 1$.

7. (Currently Amended) The method of claim 5, wherein the ~~second~~ computing the third, the fourth and the fifth points comprises:

computing an intermediate value, d, wherein d = the received point – the second point;

setting the third point = the second point + Cx * sign(dx) * 2;

setting the fourth point = the second point + i * Cy * sign(dy) * 2; and

setting the fifth point = the second point + 2(Cx * sign(dx) + i * Cy * sign(dy)),

______wherein Cx and Cy are values specifying if the two-dimensional components of the received point lie inside a boundary of the square and dx and dy are two-dimensional components of d.

8. (Currently Amended) The method of claim 5, wherein the ~~third~~ computing <u>the distances</u> comprises computing a Euclidean distance from the received point to each of the second, <u>the</u> third, <u>the</u> fourth, and <u>the</u> fifth points.

9-22. (Canceled)

23. (Currently Amended) The method of claim 1, ~~wherein the method can be used to decode a~~ <u>further comprising decoding the</u> received point in a communications system.

24. (Original) The method of claim 23, wherein the communications system is an asymmetric digital subscriber line (ADSL) compliant system.

25-27. (Canceled)

28. (New) A method for computing distances between a received point for a tone received in a receiver and a plurality of coset points in a two-dimensional grid with a constellation representing a number of bits greater than three, the method comprising:

determining a received grid point on a grid nearest to the received point;

determining whether the received grid point is valid;

determining an alternate grid point inside a specified area if the received grid point is determined to be not valid;

assigning the received grid point to a first coset point of the plurality of coset points if the received grid point is valid;

assigning the alternate grid point to the first coset point of the plurality of coset points if the received grid point is not valid;

computing remaining coset points of the plurality of coset points within the specified area, wherein each coset point is a member of a unique coset and is the closest point in its coset to the received point;

computing distances from the received point to each coset point of the plurality of coset points; and

providing the computed distances to a decoder.

29. (New) The method of claim 28, wherein the determining whether the received grid point is valid comprises determining whether the received grid point is within the constellation.

30. (New) The method of claim 28, wherein the received grid point is determined by evaluating:

$$\mathrm{round}((Rx + iRy - 1 - i)/2)*2 + 1 + I,$$

wherein Rx and Ry are two-dimensional components of the received point, I is an imaginary number, and round(.) is an operator that returns an integer number closest to a value provided to it.

31. (New) The method of claim 28, wherein the number of bits is an even value, wherein the received point can be expressed in two-dimensional components Rx and Ry, and wherein the determining whether the received grid point is valid comprises determining if Rx and Ry lie inside a square specified by the number of bits.

32. (New) The method of claim 31, further comprising computing two-dimensional components of the first coset point of the plurality of coset points based on whether the received grid point is valid, wherein, if the received grid point is valid, the received grid point is computed and, if the received grid point is not valid, the alternate grid point is computed.

33. (New) The method of claim 32, wherein determining if Rx and Ry lie inside the square comprises:

setting Cx = 1 if Rx lies inside a boundary of the square, else Cx = -1; and

setting Cy = 1 if Ry lies inside the boundary of the square, else Cy = -1,

wherein computing the two-dimensional components comprises

setting Ax = sign(RGx) * MAX_{XY} if Cx = -1, else Ax = RGx, and

setting Ay = sign(RGy) * MAX_{XY} if Cy = -1, else Ay = RGy,

wherein Ax and Ay are the two-dimensional components of the first coset point of the plurality of coset points, RGx and RGy are two-dimensional components of the received grid point if the received grid point is valid or of the alternate grid point if the received grid point is not valid, and MAX_{XY} is a value describing a size of the square and can be computed by $2^{\text{number of bits}/2} - 1$.

34. (New) The method of claim 32, wherein computing the remaining coset points comprises:

computing an intermediate value, d, wherein d = the received point – the first coset point of the plurality of coset points; and

setting the second coset point of the plurality of coset points = the first coset point of the plurality of coset points + Cx * sign(dx) * 2,

wherein Cx and Cy are values specifying if the two-dimensional components Rx and Ry of the received point lie inside a boundary of the square and dx and dy are two-dimensional components of d.

35. (New) The method of claim 34, wherein the computing the remaining coset points further comprises setting the third coset point of the plurality of coset points equal to the first coset point of the plurality of coset points + i * Cy * sign(dy) * 2.

36. (New) The method of claim 35, wherein the computing the remaining coset points further comprises setting the fourth coset point of the plurality of coset points equal to the first coset point of the plurality of coset points + 2(Cx * sign(dx) + i * Cy * sign(dy)).

37. (New) The method of claim 32, wherein the computing the distances from the received point to each coset point of the plurality of coset points comprises computing a Euclidean distance from the received point to each coset point of the plurality of coset points.

38. (New) The method of claim 28, further comprising decoding the received point in a communications system.

39. (New) The method of claim 38, wherein the communications system is an asymmetric digital subscriber line (ADSL) compliant system.